Numerical Methods week 2

Learning outcomes:

- Recall methods of solution of inhomogeneous systems of linear equations.
- Elimination methods, Gauss elimination, Gauss-Jordan elimination .
- Implement and use Gauss-Jordan Elimination to solve systems of equations.

Reading:

- Introduction to Part 3 and chapter 9 of Chapra and Canale.

Solving a system of Equations

Suppose we want to solve a system of equations $Ax = b$

Where A is a matrix of coefficients of our vector of unknowns, x and b is a vector of constants.

Explicitly this can be written (for a set of 4 equations with 4 unknowns) as:

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Augmented Matrix

It can be useful to rewrite this as an augmented matrix (https://en.wikipedia.org/wiki/Augmented_matrix)

$$\left(\begin{array}{cccc|c} a_{00} & a_{01} & a_{02} & a_{03} & b_0 \\ a_{10} & a_{11} & a_{12} & a_{13} & b_1 \\ a_{20} & a_{21} & a_{22} & a_{23} & b_2 \\ a_{30} & a_{31} & a_{32} & a_{33} & b_3 \end{array}\right)$$

Gaussian Elimination

Triangularization

We reduce the augmented matrix to row echelon form (https://en.wikipedia.org/wiki/Row_echelon_form). Let us take an initial augmented matrix as:

$$\left(\begin{array}{cccc|c} 2 & 2 & 4 & -2 & 10 \\ 1 & 3 & 2 & 4 & 17 \\ 3 & 1 & 3 & 1 & 18 \\ 1 & 3 & 4 & 2 & 27 \end{array}\right)$$

pivoting around row 0, we remove all entries below the diagonal entry in column 0, Matrix after pivoting around row 0

$$\left(\begin{array}{cccc|c} 2 & 2 & 4 & -2 & 10 \\ 0 & 2 & 0 & 5 & 12 \\ 0 & -2 & -3 & 4 & 3 \\ 0 & 2 & 2 & 3 & 22 \end{array}\right)$$

Then pivoting around row 1 we remove elements below the diagonal in column 1,

$$\left(\begin{array}{cccc|c} 2 & 2 & 4 & -2 & 10 \\ 0 & 2 & 0 & 5 & 12 \\ 0 & 0 & -3 & 9 & 15 \\ 0 & 0 & 2 & -2 & 10 \end{array}\right)$$

pivoting around row 2

Gaussian Elimination

Back substitution

From the triangularized augmented matrix we can then solve for x

$$\left(\begin{array}{cccc|c} 2 & 2 & 4 & -2 & 10 \\ 0 & 2 & 0 & 5 & 12 \\ 0 & 0 & -3 & 9 & 15 \\ 0 & 0 & 0 & 4 & 20 \end{array}\right)$$

Starting at row 3, only the coefficient of x_3 is non-zero. Converting the augmented notation to real equations using the last row we have

$$0 \cdot x_0 + 0 \cdot x_1 + 0 \cdot x_2 + 4 \cdot x_3 = 20 \implies x_3 = 5$$

Then we can work up the rows

$$\begin{aligned} 0 \cdot x_0 + 0 \cdot x_1 + -3 \cdot x_2 + 9 \cdot x_3 &= 15 \\ 0 \cdot x_0 + 0 \cdot x_1 + -3 \cdot x_2 + 9 \cdot 5 &= 15 \implies x_2 = 10 \end{aligned}$$

Now row 2

$$0 \cdot x_0 + 2 \cdot x_1 + 0 \cdot x_2 + 5 \cdot x_3 = 12$$

$$0 \cdot x_0 + 2 \cdot x_1 + 0 \cdot 10 + 5 \cdot 5 = 12 \implies x_1 = -6.5$$

And finally

$$2 \cdot x_0 + 2 \cdot -6.5 + 4 \cdot 10 + -2 \cdot 5 = 10 \implies x_0 = -3.5$$

Python Gauss Elimination

Ideally I would like everyone to be able to use C++ or python. Here is a python code.

You can run python code using spyder, a python shell or jupyter notebooks. All are available on the boxes in INB 2305 and are freely available online.

In [3]:

```python
import numpy as np # import our numpy library and call it np for short

N = 3 # size of the problem
a = np.random.uniform(1,10,(N,N+1)) # generate a N x(N+1) matrix
x = np.zeros(3) # create a solution vector of zeros ready to be filled
print(a)
```

```
[[5.43276602 3.62986307 2.23184671 4.78447635]
 [1.4933043  9.46355388 9.56326641 7.5757791 ]
 [5.05345723 1.17116046 6.01208947 6.45622803]]
```

In [4]:

```
# eliminating - we are going to modify a as we do it
for i in range(N-1): # go across the columns 1 at a time
    for j in range(i+1,N,1): # now we go down the column below the diagonal
        coeff = a[j,i]/a[i,i] # find ratio of diagonal element to value in this row
        for k in range(i,N+1,1): # no move along the row
            a[j,k] -= a[i,k]*coeff # subtract the right amoung of the row on the diagona
l
print(a)
```

```
[[ 5.43276602e+00  3.62986307e+00  2.23184671e+00  4.78447635e+00]
 [-2.22044605e-16  8.46581349e+00  8.94979875e+00  6.26067018e+00]
 [ 0.00000000e+00  0.00000000e+00  6.26741157e+00  3.63664742e+00]]
```

In [5]:

```
# now back substitute
for i in range(N-1,-1,-1): # start at the bottom and work up
    x[i] = a[i,N+1-1]
    for j in range(i+1, N,1):
        x[i] -= a[i,j]*x[j]
    x[i] /= a[i,i]
print(x)
```

```
[0.55804216 0.12610429 0.5802471 ]
```

To make things more usable we could put the Guassian Elimination code into a function:

In [6]:

```python
import numpy as np

def gauss_elim(a, N):
    """solves a set of linear equations
    N must be the number of rows (dimension of solution).
    Warning: On exit the original matrix is changed to row echelon form.
    """
    x = np.zeros(N);
    # eliminating
    for i in range(N-1):
        for j in range(i+1,N,1):
            coeff = a[j,i]/a[i,i]
            for k in range(i,N+1,1):
                a[j,k] -= a[i,k]*coeff
    # now back substitute
    for i in range(N-1,-1,-1):
        x[i] = a[i,N+1-1]
        for j in range(i+1, N,1):
            x[i] -= a[i,j]*x[j]
        x[i] /= a[i,i]
    return x

# build a random 2d 3x4 augmented matrix
x = gauss_elim(a,N)
print(x)
```

```
[0.55804216 0.12610429 0.5802471 ]
```

Gauss Elimination

Use the code to find the solutions of the following systems

$$
\begin{aligned}
3x_0 + 4x_1 - 7x_2 &= 23 \\
7x_0 - x_1 + 2x_2 &= 14 \\
x_0 + 10x_1 - 2x_2 &= 33
\end{aligned}
$$

Can you find the solutions to this system of equations? Why not?

$$
\begin{aligned}
1x_0 + 2x_1 + 3x_2 &= 1 \\
4x_0 + 5x_1 + 6x_2 &= 2 \\
7x_0 + 8x_1 + 9x_2 &= 3
\end{aligned}
$$

Guass-Jordan elimination

Guass-Jordan elimination is very similar to Gauss elimination. Instead of triangularization, we make a completely diagonal matrix. Or more exactly we reduce the augmented matrix to $\mathbf{reduced}$ row echelon form. Initial matrix is:

$$\left(\begin{array}{cccc|c} 2 & 2 & 4 & -2 & 10 \\ 1 & 3 & 2 & 4 & 17 \\ 3 & 1 & 3 & 1 & 18 \\ 1 & 3 & 4 & 2 & 27 \end{array}\right)$$

as before

The first step is the same: pivoting around row 0

$$\left(\begin{array}{cccc|c} 2 & 2 & 4 & -2 & 10 \\ 0 & 2 & 0 & 5 & 12 \\ 0 & -2 & -3 & 4 & 3 \\ 0 & 2 & 2 & 3 & 22 \end{array}\right)$$

The first step is the same:
pivoting around row 0

$$\left(\begin{array}{cccc|c} 2 & 2 & 4 & -2 & 10 \\ 0 & 2 & 0 & 5 & 12 \\ 0 & -2 & -3 & 4 & 3 \\ 0 & 2 & 2 & 3 & 22 \end{array}\right)$$

But now, pivoting around row 1, we remove entries above $\mathbf{and}$ below the diagonal of column 1

$$\left(\begin{array}{cccc|c} 2 & 0 & 4 & -7 & -2 \\ 0 & 2 & 0 & 5 & 12 \\ 0 & 0 & -3 & 9 & 15 \\ 0 & 0 & 2 & -2 & 10 \end{array}\right)$$

This continues pivoting around row 2

And finally, pivoting around row 3

$$\left(\begin{array}{cccc|c} 2 & 0 & 0 & 0 & -7 \\ 0 & 2 & 0 & 0 & -13 \\ 0 & 0 & -3 & 0 & -30 \\ 0 & 0 & 0 & 4 & 20 \end{array}\right)$$

We can then divide each row by the final coefficients to get:

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & -3.5 \\ 0 & 1 & 0 & 0 & -6.5 \\ -0 & -0 & 1 & -0 & 10 \\ 0 & 0 & 0 & 1 & 5 \end{array}\right)$$

And we can just read off the solutions, x.

Gauss-Jordan Elimination for Matrix Inversion

We can solve the equation $AB = I$ using exactly the same method:

$$\left(\begin{array}{ccc|ccc} 2 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 2 & -1 & 2 & 0 & 0 & 1 \end{array}\right)$$

pivoting around row 0

$$\left(\begin{array}{ccc|ccc} 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & -0.5 & -1.5 & -0.5 & 1 & 0 \\ 0 & -2 & 1 & -1 & 0 & 1 \end{array}\right)$$

pivoting around row 1

$$\left(\begin{array}{ccc|ccc} 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & -0.5 & -1.5 & -0.5 & 1 & 0 \\ 0 & 0 & 7 & 1 & -4 & 1 \end{array}\right)$$

pivoting around row 2

$$\left(\begin{array}{ccc|ccc} 2 & 0 & 0 & 0.285714 & 0.857143 & 0.285714 \\ 0 & -0.5 & 0 & -0.285714 & 0.142857 & 0.214286 \\ 0 & 0 & 7 & 1 & -4 & 1 \end{array}\right)$$

Scaling the rows, the final matrix is:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.142857 & 0.428571 & 0.142857 \\ -0 & 1 & -0 & 0.571429 & -0.285714 & -0.428571 \\ 0 & 0 & 1 & 0.142857 & -0.571429 & 0.142857 \end{array}\right)$$

The RHS of the augmented matrix is A^{-1}

Gauss-Jordan Elimination

Alter the code for Gauss elimination to instead perform Gauss-Jordan elimination.

- I suggest copying your file for now, renaming rather than altering the previous code directly. - You should change the second loop so that it goes over all rows.
- You should add an `if` statement to skip the row with the same index as the column you are working on.
- When the matrix is diagonal, divide each row by the value of the remaining diagonal element to get the identity matrix and x.
- Test regularly as you make the alterations.

Solve the same matrix problems as before and check the result is the same.

Gauss-Jordan Matrix inversion

Make a new Gauss-Jordan routine that can calculate the inverse of a matrix.

- You need to extend the number of columns in the augmented matrix
- You need to extend range of the loops that go over the columns

Find the inverse of the matrix

$$\begin{pmatrix} 2 & 2 & 4 & -2 \\ 1 & 3 & 2 & 4 \\ 3 & 1 & 3 & 1 \\ 1 & 3 & 4 & 2 \end{pmatrix}$$

Check your solution is correct by multiplying the original and inverse matrices.

Test yourself

Try the test in the assessments tab of Bb. It is based on last week's material. The difficulty will probably be working with the matrices. Try to get a feel for the indices and accessing individual elements.

Summary and Further Reading

You should be reading additional material to provide a solid background to what we do in class

All the textbooks in the book list on Bb contain sections on solving linear equations. I suggest Chapter 9 of Chapra and Canale for starters.

Homework

Before next week read about extra steps that can be performed to improve elimination methods.

Read about LU decomposition of square matrices, Chapter 10 of Chapra and Canale.